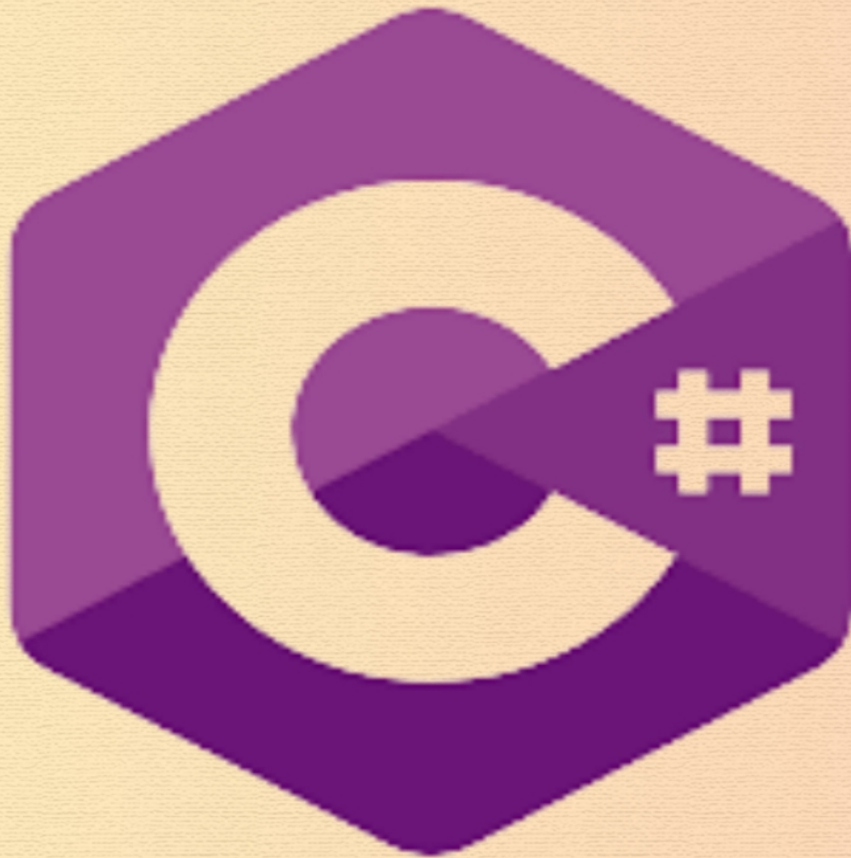


2023 NEW EDITION
FOR BEGINNERS



100 PROGRAMS

BEST FOR COMPUTER SCIENCE STUDENTS

DEVBRAT RUDRA

C#

100 PROGRAM

By – Devbrat Rudra

www.rudyhelp.com

(NOTE: PAGE NUMBER 100 – HOW TO RUN C# PROGRAM)

SET - 1

1. Hello World:

```
using System;

class HelloWorld {
    static void Main() {
        Console.WriteLine("Hello, World!");
    }
}
```

2. Simple Calculator:

```
using System;

class Calculator {
    static void Main() {
        Console.WriteLine("Enter the first number: ");
        double num1 = Convert.ToDouble(Console.ReadLine());
```

```

        Console.WriteLine("Enter the second number: ");
        double num2 = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Sum: " + (num1 + num2));
    }
}

```

3. Factorial of a Number:

```

using System;

class Factorial {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        int factorial = 1;
        for (int i = 1; i <= num; i++) {
            factorial *= i;
        }

        Console.WriteLine("Factorial: " + factorial);
    }
}

```

4. Fibonacci Series:

```

using System;

class Fibonacci {
    static void Main() {
        Console.WriteLine("Enter the number of terms: ");
        int n = Convert.ToInt32(Console.ReadLine());
    }
}

```

```

int a = 0, b = 1, c;
Console.WriteLine("Fibonacci Series: ");
for (int i = 0; i < n; i++) {
    Console.WriteLine(a + " ");
    c = a + b;
    a = b;
    b = c;
}
}
}

```

5. Check Whether a Number is Prime:

using System;

```

class PrimeNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        bool isPrime = true;
        for (int i = 2; i <= num / 2; i++) {
            if (num % i == 0) {
                isPrime = false;
                break;
            }
        }

        if (isPrime)
            Console.WriteLine("Prime");
        else
            Console.WriteLine("Not Prime");
    }
}

```

```
}  
}
```

6. Simple Interest Calculator:

using System;

```
class SimpleInterest {  
    static void Main() {  
        Console.WriteLine("Enter principal amount: ");  
        double principal = Convert.ToDouble(Console.ReadLine());  
  
        Console.WriteLine("Enter rate of interest: ");  
        double rate = Convert.ToDouble(Console.ReadLine());  
  
        Console.WriteLine("Enter time in years: ");  
        double time = Convert.ToDouble(Console.ReadLine());  
  
        double interest = (principal * rate * time) / 100;  
        Console.WriteLine("Simple Interest: " + interest);  
    }  
}
```

7. Check Whether a Number is Even or Odd:

using System;

```
class EvenOdd {  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
  
        if (num % 2 == 0)  
            Console.WriteLine("Even");  
    }  
}
```

```

        else
            Console.WriteLine("Odd");
    }
}

```

8. Print the Sum of Natural Numbers:

using System;

```

class SumOfNaturalNumbers {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        int sum = 0;
        for (int i = 1; i <= num; i++) {
            sum += i;
        }

        Console.WriteLine("Sum of natural numbers: " + sum);
    }
}

```

9. Reverse a String:

using System;

```

class StringReverse {
    static void Main() {
        Console.WriteLine("Enter a string: ");
        string input = Console.ReadLine();

        char[] charArray = input.ToCharArray();
    }
}

```

```

        Array.Reverse(charArray);
        string reversedString = new string(charArray);

        Console.WriteLine("Reversed string: " + reversedString);
    }
}

```

10. Find the Maximum and Minimum Numbers in an Array:

using System;

```

class MaxMinInArray {
    static void Main() {
        int[] numbers = { 5, 2, 8, 1, 9, 4, 6 };

        int max = numbers[0];
        int min = numbers[0];

        foreach (int number in numbers) {
            if (number > max)
                max = number;

            if (number < min)
                min = number;
        }

        Console.WriteLine("Maximum: " + max);
        Console.WriteLine("Minimum: " + min);
    }
}

```

SET – 2

1. Check Leap Year:

using System;

```
class LeapYear {  
    static void Main() {  
        Console.WriteLine("Enter a year: ");  
        int year = Convert.ToInt32(Console.ReadLine());  
  
        if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0)  
            Console.WriteLine("Leap Year");  
        else  
            Console.WriteLine("Not a Leap Year");  
    }  
}
```

2. Reverse an Array:

using System;

```
class ArrayReverse {  
    static void Main() {  
        int[] arr = { 1, 2, 3, 4, 5 };  
        Array.Reverse(arr);  
  
        Console.Write("Reversed Array: ");  
        foreach (int element in arr) {  
            Console.Write(element + " ");  
        }  
    }  
}
```

3. Palindrome Check:


```

using System;

class Palindrome {
    static void Main() {
        Console.WriteLine("Enter a string: ");
        string input = Console.ReadLine();

        char[] charArray = input.ToCharArray();
        Array.Reverse(charArray);
        string reversedString = new string(charArray);

        if (input.Equals(reversedString,
StringComparison.OrdinalIgnoreCase))
            Console.WriteLine("Palindrome");
        else
            Console.WriteLine("Not a Palindrome");
    }
}

```

4. Print ASCII Value of a Character:

```

using System;

class ASCIIValue {
    static void Main() {
        Console.WriteLine("Enter a character: ");
        char ch = Convert.ToChar(Console.ReadLine());

        int asciiValue = (int)ch;
        Console.WriteLine("ASCII Value: " + asciiValue);
    }
}

```

5. Generate Random Numbers:

using System;

```
class RandomNumbers {  
    static void Main() {  
        Random random = new Random();  
        Console.WriteLine("Random Numbers: ");  
  
        for (int i = 0; i < 5; i++) {  
            Console.WriteLine(random.Next(1, 100));  
        }  
    }  
}
```

6. Check Armstrong Number:

using System;

```
class ArmstrongNumber {  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
  
        int originalNumber, remainder, result = 0;  
        originalNumber = num;  
  
        while (originalNumber != 0) {  
            remainder = originalNumber % 10;  
            result += remainder * remainder * remainder;  
            originalNumber /= 10;  
        }  
  
        if (result == num)
```

```

        Console.WriteLine("Armstrong Number");
    else
        Console.WriteLine("Not an Armstrong Number");
    }
}

```

7. Bubble Sort:

```

using System;

class BubbleSort {
    static void Main() {
        int[] arr = { 64, 34, 25, 12, 22, 11, 90 };

        for (int i = 0; i < arr.Length - 1; i++) {
            for (int j = 0; j < arr.Length - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }

        Console.Write("Sorted Array: ");
        foreach (int element in arr) {
            Console.Write(element + " ");
        }
    }
}

```

8. Binary Search:

```
using System;
```

```
class BinarySearch {  
    static int Search(int[] arr, int x) {  
        int left = 0, right = arr.Length - 1;  
        while (left <= right) {  
            int mid = left + (right - left) / 2;  
  
            if (arr[mid] == x)  
                return mid;  
  
            if (arr[mid] < x)  
                left = mid + 1;  
            else  
                right = mid - 1;  
        }  
  
        return -1;  
    }  
  
    static void Main() {  
        int[] arr = { 2, 3, 4, 10, 40 };  
        int x = 10;  
        int result = Search(arr, x);  
  
        if (result == -1)  
            Console.WriteLine("Element not present");  
        else  
            Console.WriteLine("Element found at index " + result);  
    }  
}
```

9. Check Whether a String is Anagram:

```

using System;

class AnagramCheck {
    static void Main() {
        Console.WriteLine("Enter the first string: ");
        string str1 = Console.ReadLine();

        Console.WriteLine("Enter the second string: ");
        string str2 = Console.ReadLine();

        char[] charArray1 = str1.ToLower().ToCharArray();
        char[] charArray2 = str2.ToLower().ToCharArray();

        Array.Sort(charArray1);
        Array.Sort(charArray2);

        string sortedStr1 = new string(charArray1);
        string sortedStr2 = new string(charArray2);

        if (sortedStr1.Equals(sortedStr2))
            Console.WriteLine("Anagrams");
        else
            Console.WriteLine("Not Anagrams");
    }
}

```

10. Print the ASCII Table:

```

using System;

class ASCIITable {
    static void Main() {
        Console.WriteLine("ASCII Table: ");
        for (int i = 0; i < 256; i++) {

```

```

        Console.WriteLine("ASCII Value: " + i + ", Character: " +
(char)i);
    }
}
}

```

SET – 3

1. Check for Armstrong Numbers in a Range:

using System;

```

class ArmstrongNumbersInRange {
    static void Main() {
        Console.WriteLine("Enter the starting number: ");
        int start = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the ending number: ");
        int end = Convert.ToInt32(Console.ReadLine());

        for (int i = start; i <= end; i++) {
            int num = i;
            int originalNumber, remainder, result = 0;
            originalNumber = num;

            while (originalNumber != 0) {
                remainder = originalNumber % 10;
                result += remainder * remainder * remainder;
                originalNumber /= 10;
            }

            if (result == num)

```

```

        Console.WriteLine(num + " is an Armstrong Number");
    }
}
}

```

2. Check for Perfect Numbers in a Range:

using System;

```

class PerfectNumbersInRange {
    static void Main() {
        Console.WriteLine("Enter the starting number: ");
        int start = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the ending number: ");
        int end = Convert.ToInt32(Console.ReadLine());

        for (int i = start; i <= end; i++) {
            int num = i;
            int sum = 0;
            for (int j = 1; j < num; j++) {
                if (num % j == 0)
                    sum += j;
            }

            if (sum == num)
                Console.WriteLine(num + " is a Perfect Number");
        }
    }
}

```

3. Print the Multiplication Table:

```

using System;

class MultiplicationTable {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Multiplication Table: ");
        for (int i = 1; i <= 10; i++) {
            Console.WriteLine(num + " * " + i + " = " + (num * i));
        }
    }
}

```

4. Implement Linear Search:

```

using System;

class LinearSearch {
    static int Search(int[] arr, int x) {
        for (int i = 0; i < arr.Length; i++) {
            if (arr[i] == x)
                return i;
        }
        return -1;
    }

    static void Main() {
        int[] arr = { 2, 3, 4, 10, 40 };
        int x = 10;
        int result = Search(arr, x);

        if (result == -1)

```



```

        Console.WriteLine("Element not present");
    else
        Console.WriteLine("Element found at index " + result);
    }
}

```

5. Find the GCD and LCM of Two Numbers:

using System;

```

class GCDandLCM {
    static int FindGCD(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }

    static int FindLCM(int a, int b) {
        return (a * b) / FindGCD(a, b);
    }

    static void Main() {
        Console.WriteLine("Enter two numbers: ");
        int num1 = Convert.ToInt32(Console.ReadLine());
        int num2 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("GCD: " + FindGCD(num1, num2));
        Console.WriteLine("LCM: " + FindLCM(num1, num2));
    }
}

```

6. Print the Nth Fibonacci Number:

using System;

```
class NthFibonacci {  
    static int Fibonacci(int n) {  
        if (n <= 1)  
            return n;  
        return Fibonacci(n - 1) + Fibonacci(n - 2);  
    }  
  
    static void Main() {  
        Console.WriteLine("Enter the value of n: ");  
        int n = Convert.ToInt32(Console.ReadLine());  
  
        Console.WriteLine("Nth Fibonacci number: " + Fibonacci(n));  
    }  
}
```

7. Generate a Simple Pattern:

using System;

```
class Pattern {  
    static void Main() {  
        Console.WriteLine("Enter the number of rows: ");  
        int rows = Convert.ToInt32(Console.ReadLine());  
  
        for (int i = 1; i <= rows; i++) {  
            for (int j = 1; j <= i; j++) {  
                Console.Write("* ");  
            }  
            Console.WriteLine();  
        }  
    }  
}
```

```
}  
}
```

8. Print the Power of 2:

using System;

```
class PowerOf2 {  
    static void Main() {  
        Console.WriteLine("Enter the value of n: ");  
        int n = Convert.ToInt32(Console.ReadLine());  
  
        for (int i = 0; i <= n; i++) {  
            Console.WriteLine("2 ^ " + i + " = " + Math.Pow(2, i));  
        }  
    }  
}
```

9. Calculate the Area of Different Geometric Shapes:

using System;

```
class AreaCalculator {  
    static void Main() {  
        Console.WriteLine("Choose a shape - 1. Rectangle, 2. Circle: ");  
        int choice = Convert.ToInt32(Console.ReadLine());  
  
        switch (choice) {  
            case 1:  
                Console.WriteLine("Enter length: ");  
                double length = Convert.ToDouble(Console.ReadLine());  
                Console.WriteLine("Enter breadth: ");  
                double breadth = Convert.ToDouble(Console.ReadLine());
```

```

        Console.WriteLine("Area of Rectangle: " + (length *
breadth));
        break;
    case 2:
        Console.WriteLine("Enter radius: ");
        double radius = Convert.ToDouble(Console.ReadLine());
        Console.WriteLine("Area of Circle: " + (Math.PI * radius *
radius));
        break;
    default:
        Console.WriteLine("Invalid Choice");
        break;
    }
}
}
}

```

10. Convert Binary to Decimal and Decimal to Binary:

using System;

```

class Conversion {
    static void Main() {
        Console.WriteLine("Choose an option - 1. Binary to Decimal, 2.
Decimal to Binary: ");
        int choice = Convert.ToInt32(Console.ReadLine());

        switch (choice) {
            case 1:
                Console.WriteLine("Enter a binary number: ");
                string binary = Console.ReadLine();
                int decimalValue = Convert.ToInt32(binary, 2);
                Console.WriteLine("Decimal Value: " + decimalValue);
            }
        }
    }
}

```

```

        break;
    case 2:
        Console.WriteLine("Enter a decimal number: ");
        int decimalNum = Convert.ToInt32(Console.ReadLine());
        string binaryValue = Convert.ToString(decimalNum, 2);
        Console.WriteLine("Binary Value: " + binaryValue);
        break;
    default:
        Console.WriteLine("Invalid Choice");
        break;
    }
}
}

```

SET – 4

1. Count the Number of Digits in an Integer:

using System;

```

class CountDigits {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        int count = 0;
        while (num != 0) {
            num /= 10;
            count++;
        }

        Console.WriteLine("Number of digits: " + count);
    }
}

```

```
}  
}
```

2. Check for Perfect Square:

using System;

```
class PerfectSquareCheck {  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
  
        int sqrt = (int)Math.Sqrt(num);  
        if (sqrt * sqrt == num)  
            Console.WriteLine("Perfect Square");  
        else  
            Console.WriteLine("Not a Perfect Square");  
    }  
}
```

3. Check for Palindrome Number:

using System;

```
class PalindromeNumber {  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
  
        int originalNumber = num;  
        int reversedNumber = 0;  
  
        while (num > 0) {
```

```

        int remainder = num % 10;
        reversedNumber = reversedNumber * 10 + remainder;
        num /= 10;
    }

    if (originalNumber == reversedNumber)
        Console.WriteLine("Palindrome Number");
    else
        Console.WriteLine("Not a Palindrome Number");
    }
}

```

4. Implement Selection Sort:

```

using System;

class SelectionSort {
    static void Main() {
        int[] arr = { 64, 25, 12, 22, 11 };

        for (int i = 0; i < arr.Length - 1; i++) {
            int minIndex = i;
            for (int j = i + 1; j < arr.Length; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }

        Console.WriteLine("Sorted Array: ");
    }
}

```

```

        foreach (int element in arr) {
            Console.Write(element + " ");
        }
    }
}

```

5. Generate a Simple Pyramid Pattern:

using System;

```

class PyramidPattern {
    static void Main() {
        Console.WriteLine("Enter the number of rows: ");
        int rows = Convert.ToInt32(Console.ReadLine());

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows - i; j++) {
                Console.Write(" ");
            }
            for (int k = 1; k <= 2 * i - 1; k++) {
                Console.Write("*");
            }
            Console.WriteLine();
        }
    }
}

```

6. Calculate the Area and Perimeter of a Rectangle:

using System;

```

class Rectangle {

```



```

static void Main() {
    Console.WriteLine("Enter length: ");
    double length = Convert.ToDouble(Console.ReadLine());

    Console.WriteLine("Enter breadth: ");
    double breadth = Convert.ToDouble(Console.ReadLine());

    double area = length * breadth;
    double perimeter = 2 * (length + breadth);

    Console.WriteLine("Area: " + area);
    Console.WriteLine("Perimeter: " + perimeter);
}
}

```

7. Reverse a Number:

```

using System;

class NumberReverse {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        int reversedNumber = 0;
        while (num > 0) {
            int remainder = num % 10;
            reversedNumber = reversedNumber * 10 + remainder;
            num /= 10;
        }

        Console.WriteLine("Reversed Number: " + reversedNumber);
    }
}

```

8. Print Pascal's Triangle:

using System;

```
class PascalTriangle {
    static void Main() {
        Console.WriteLine("Enter the number of rows: ");
        int rows = Convert.ToInt32(Console.ReadLine());

        for (int i = 0; i < rows; i++) {
            int number = 1;
            Console.Write(new string(' ', rows - i));

            for (int j = 0; j <= i; j++) {
                Console.Write(number + " ");
                number = number * (i - j) / (j + 1);
            }

            Console.WriteLine();
        }
    }
}
```

9. Calculate the Power of a Number:

using System;

```
class Power {
    static void Main() {
        Console.WriteLine("Enter the base: ");
        double baseNum = Convert.ToDouble(Console.ReadLine());

        Console.WriteLine("Enter the exponent: ");
        int exponent = Convert.ToInt32(Console.ReadLine());
```

```

        double result = Math.Pow(baseNum, exponent);
        Console.WriteLine("Result: " + result);
    }
}

```

10. Check for Disarium Number:

using System;

```

class DisariumNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        string numString = num.ToString();
        int sum = 0;

        for (int i = 0; i < numString.Length; i++) {
            sum += (int)Math.Pow(int.Parse(numString[i].ToString()), i +
1);
        }

        if (sum == num)
            Console.WriteLine("Disarium Number");
        else
            Console.WriteLine("Not a Disarium Number");
    }
}

```

SET – 5

1. Check for Strong Number:

```
using System;
```

```
class StrongNumber {  
    static int Factorial(int num) {  
        if (num == 0 || num == 1)  
            return 1;  
        return num * Factorial(num - 1);  
    }  
  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
        int originalNum = num;  
        int sum = 0;  
  
        while (num > 0) {  
            int digit = num % 10;  
            sum += Factorial(digit);  
            num /= 10;  
        }  
  
        if (sum == originalNum)  
            Console.WriteLine("Strong Number");  
        else  
            Console.WriteLine("Not a Strong Number");  
    }  
}
```

2. Calculate the Circumference and Area of a Circle:

```
using System;
```

```
class Circle {
```

```

static void Main() {
    Console.WriteLine("Enter the radius: ");
    double radius = Convert.ToDouble(Console.ReadLine());

    double circumference = 2 * Math.PI * radius;
    double area = Math.PI * radius * radius;

    Console.WriteLine("Circumference: " + circumference);
    Console.WriteLine("Area: " + area);
}
}

```

3. Convert Temperature from Celsius to Fahrenheit and Vice Versa:

using System;

```

class TemperatureConverter {
    static void Main() {
        Console.WriteLine("Choose an option - 1. Celsius to Fahrenheit,
2. Fahrenheit to Celsius: ");
        int choice = Convert.ToInt32(Console.ReadLine());

        switch (choice) {
            case 1:
                Console.WriteLine("Enter temperature in Celsius: ");
                double celsius = Convert.ToDouble(Console.ReadLine());
                double fahrenheit = (celsius * 9 / 5) + 32;
                Console.WriteLine("Temperature in Fahrenheit: " +
fahrenheit);
                break;
            case 2:
                Console.WriteLine("Enter temperature in Fahrenheit: ");

```

```

        double fahrenheit2 =
Convert.ToDouble(Console.ReadLine());
        double celsius2 = (fahrenheit2 - 32) * 5 / 9;
        Console.WriteLine("Temperature in Celsius: " + celsius2);
        break;
    default:
        Console.WriteLine("Invalid Choice");
        break;
    }
}
}

```

4. Find the HCF and LCM of Two Numbers:

using System;

```

class HCFandLCM {
    static int FindHCF(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }

    static int FindLCM(int a, int b) {
        return (a * b) / FindHCF(a, b);
    }

    static void Main() {
        Console.WriteLine("Enter two numbers: ");
    }
}

```

```

        int num1 = Convert.ToInt32(Console.ReadLine());
        int num2 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("HCF: " + FindHCF(num1, num2));
        Console.WriteLine("LCM: " + FindLCM(num1, num2));
    }
}

```

5. Implement Insertion Sort:

```

using System;

class InsertionSort {
    static void Main() {
        int[] arr = { 12, 11, 13, 5, 6 };

        for (int i = 1; i < arr.Length; i++) {
            int key = arr[i];
            int j = i - 1;

            while (j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }

        Console.Write("Sorted Array: ");
        foreach (int element in arr) {
            Console.Write(element + " ");
        }
    }
}

```

6. Check for Harshad Number:

using System;

```
class HarshadNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int originalNum = num;
        int sum = 0;

        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }

        if (originalNum % sum == 0)
            Console.WriteLine("Harshad Number");
        else
            Console.WriteLine("Not a Harshad Number");
    }
}
```

7. Generate the First N Prime Numbers:

using System;

```
class PrimeNumbers {
    static bool IsPrime(int num) {
        if (num <= 1)
            return false;

        for (int i = 2; i <= Math.Sqrt(num); i++) {
            if (num % i == 0)
```



```

        return false;
    }
    return true;
}

static void Main() {
    Console.WriteLine("Enter the value of n: ");
    int n = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("First " + n + " prime numbers are: ");

    int count = 0;
    int number = 2;
    while (count < n) {
        if (IsPrime(number)) {
            Console.Write(number + " ");
            count++;
        }
        number++;
    }
}

```

8. Check for Abundant Number:

```

using System;

class AbundantNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int sum = 0;

        for (int i = 1; i <= num / 2; i++) {

```

```

        if (num % i == 0)
            sum += i;
    }

    if (sum > num)
        Console.WriteLine("Abundant Number");
    else
        Console.WriteLine("Not an Abundant Number");
    }
}

```

9. Calculate the Sum of the Digits of a Number:

```

using System;

class SumOfDigits {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        int sum = 0;
        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }

        Console.WriteLine("Sum of digits: " + sum);
    }
}

```

10. Check for Pronic Number:

```

using System;

```

```

class PronicNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        bool isPronic = false;
        for (int i = 0; i <= Math.Sqrt(num); i++) {
            if (i * (i + 1) == num) {
                isPronic = true;
                break;
            }
        }

        if (isPronic)
            Console.WriteLine("Pronic Number");
        else
            Console.WriteLine("Not a Pronic Number");
    }
}

```

SET – 6

1. Check for Automorphic Number:

using System;

```

class AutomorphicNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        int square = num * num;

        string strNum = num.ToString();
    }
}

```

```

        string strSquare = square.ToString();

        if (strSquare.EndsWith(strNum))
            Console.WriteLine("Automorphic Number");
        else
            Console.WriteLine("Not an Automorphic Number");
    }
}

```

2. Implement Quick Sort:

```

using System;

class QuickSort {
    static void Sort(int[] arr, int low, int high) {
        if (low < high) {
            int pivot = Partition(arr, low, high);

            Sort(arr, low, pivot - 1);
            Sort(arr, pivot + 1, high);
        }
    }

    static int Partition(int[] arr, int low, int high) {
        int pivot = arr[high];
        int i = low - 1;

        for (int j = low; j < high; j++) {
            if (arr[j] < pivot) {
                i++;
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

```

```

        }
    }

    int temp1 = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = temp1;

    return i + 1;
}

static void Main() {
    int[] arr = { 10, 7, 8, 9, 1, 5 };
    int n = arr.Length;

    Sort(arr, 0, n - 1);

    Console.Write("Sorted array: ");
    foreach (int element in arr) {
        Console.Write(element + " ");
    }
}
}

```

3. Check for Krishnamurthy Number:

```

using System;

class KrishnamurthyNumber {
    static int Factorial(int num) {
        if (num == 0 || num == 1)
            return 1;
        return num * Factorial(num - 1);
    }
}

```

```

static void Main() {
    Console.WriteLine("Enter a number: ");
    int num = Convert.ToInt32(Console.ReadLine());
    int originalNum = num;
    int sum = 0;

    while (num > 0) {
        int digit = num % 10;
        sum += Factorial(digit);
        num /= 10;
    }

    if (sum == originalNum)
        Console.WriteLine("Krishnamurthy Number");
    else
        Console.WriteLine("Not a Krishnamurthy Number");
}
}

```

4. Check for Disarium Number in a Range:

```

using System;

class DisariumNumbersInRange {
    static void Main() {
        Console.WriteLine("Enter the starting number: ");
        int start = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the ending number: ");
        int end = Convert.ToInt32(Console.ReadLine());

        for (int i = start; i <= end; i++) {
            int num = i;

```

```

        string numString = num.ToString();
        int sum = 0;

        for (int j = 0; j < numString.Length; j++) {
            sum += (int)Math.Pow(int.Parse(numString[j].ToString()), j
+ 1);
        }

        if (sum == num)
            Console.WriteLine(num + " is a Disarium Number");
    }
}
}

```

5. Print the Pattern of a Diamond:

using System;

```

class DiamondPattern {
    static void Main() {
        Console.WriteLine("Enter the number of rows: ");
        int rows = Convert.ToInt32(Console.ReadLine());

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows - i; j++) {
                Console.Write(" ");
            }
            for (int k = 1; k <= 2 * i - 1; k++) {
                Console.Write("*");
            }
            Console.WriteLine();
        }
    }
}

```

```

    for (int i = rows - 1; i >= 1; i--) {
        for (int j = 1; j <= rows - i; j++) {
            Console.Write(" ");
        }
        for (int k = 1; k <= 2 * i - 1; k++) {
            Console.Write("*");
        }
        Console.WriteLine();
    }
}

```

6. Check for Niven Number:

using System;

```

class NivenNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int originalNum = num;
        int sum = 0;

        while (num > 0) {
            sum += num % 10;
            num /= 10;
        }

        if (originalNum % sum == 0)
            Console.WriteLine("Niven Number");
        else
            Console.WriteLine("Not a Niven Number");
    }
}

```



```
}  
}
```

7. Calculate the Average of N Numbers:

using System;

```
class Average {  
    static void Main() {  
        Console.WriteLine("Enter the value of n: ");  
        int n = Convert.ToInt32(Console.ReadLine());  
  
        int sum = 0;  
        for (int i = 0; i < n; i++) {  
            Console.WriteLine("Enter number " + (i + 1) + ": ");  
            int num = Convert.ToInt32(Console.ReadLine());  
            sum += num;  
        }  
  
        double average = (double)sum / n;  
        Console.WriteLine("Average: " + average);  
    }  
}
```

8. Print the Pattern of a Hollow Rectangle:

using System;

```
class HollowRectanglePattern {  
    static void Main() {  
        Console.WriteLine("Enter the number of rows: ");  
        int rows = Convert.ToInt32(Console.ReadLine());  
        Console.WriteLine("Enter the number of columns: ");
```

```

int columns = Convert.ToInt32(Console.ReadLine());

for (int i = 1; i <= rows; i++) {
    for (int j = 1; j <= columns; j++) {
        if (i == 1 || i == rows || j == 1 || j == columns)
            Console.Write("*");
        else
            Console.Write(" ");
    }
    Console.WriteLine();
}
}

```

9. Check for Happy Number:

using System;

```

class HappyNumber {
    static int SquareSumOfDigits(int num) {
        int sum = 0;
        while (num > 0) {
            int digit = num % 10;
            sum += digit * digit;
            num /= 10;
        }
        return sum;
    }

    static bool IsHappyNumber(int num) {
        while (num != 1 && num != 4) {
            num = SquareSumOfDigits(num);
        }
    }
}

```

```

    }
    return num == 1;
}

static void Main() {
    Console.WriteLine("Enter a number: ");
    int num = Convert.ToInt32(Console.ReadLine());

    if (IsHappyNumber(num))
        Console.WriteLine("Happy Number");
    else
        Console.WriteLine("Not a Happy Number");
}
}

```

10. Check for Evil Number:

```

using System;

class EvilNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        string binary = Convert.ToString(num, 2);
        int count = 0;

        foreach (char ch in binary) {
            if (ch == '1')
                count++;
        }

        if (count % 2 == 0)
            Console.WriteLine("Evil Number");
    }
}

```

```

        else
            Console.WriteLine("Not an Evil Number");
    }
}

```

SET – 7

1. Check for Spy Number:

using System;

```

class SpyNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int originalNum = num;
        int sum = 0;
        int product = 1;

        while (num > 0) {
            int digit = num % 10;
            sum += digit;
            product *= digit;
            num /= 10;
        }

        if (sum == product)
            Console.WriteLine("Spy Number");
        else
            Console.WriteLine("Not a Spy Number");
    }
}

```

```
}
```

2. Implement Merge Sort:

using System;

```
class MergeSort {  
    static void Merge(int[] arr, int l, int m, int r) {  
        int n1 = m - l + 1;  
        int n2 = r - m;  
  
        int[] L = new int[n1];  
        int[] R = new int[n2];  
  
        for (int i = 0; i < n1; ++i)  
            L[i] = arr[l + i];  
        for (int j = 0; j < n2; ++j)  
            R[j] = arr[m + 1 + j];  
  
        int k = l;  
        int x = 0, y = 0;  
  
        while (x < n1 && y < n2) {  
            if (L[x] <= R[y]) {  
                arr[k] = L[x];  
                x++;  
            }  
            else {  
                arr[k] = R[y];  
                y++;  
            }  
            k++;  
        }  
    }  
}
```

```

while (x < n1) {
    arr[k] = L[x];
    x++;
    k++;
}

while (y < n2) {
    arr[k] = R[y];
    y++;
    k++;
}
}

static void Sort(int[] arr, int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;

        Sort(arr, l, m);
        Sort(arr, m + 1, r);

        Merge(arr, l, m, r);
    }
}

static void Main() {
    int[] arr = { 12, 11, 13, 5, 6, 7 };
    int n = arr.Length;

    Sort(arr, 0, n - 1);

    Console.WriteLine("Sorted array: ");
    foreach (int element in arr) {
        Console.Write(element + " ");
    }
}

```

```
    }  
}
```

3. Check for Fibonacci Number:

using System;

```
class FibonacciNumber {  
    static bool IsPerfectSquare(int x) {  
        int s = (int)Math.Sqrt(x);  
        return s * s == x;  
    }  
  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
  
        int val1 = 5 * num * num + 4;  
        int val2 = 5 * num * num - 4;  
  
        if (IsPerfectSquare(val1) || IsPerfectSquare(val2))  
            Console.WriteLine("Fibonacci Number");  
        else  
            Console.WriteLine("Not a Fibonacci Number");  
    }  
}
```

4. Check for Harshad Number in a Range:

using System;

```
class HarshadNumbersInRange {  
    static void Main() {
```

```

Console.WriteLine("Enter the starting number: ");
int start = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the ending number: ");
int end = Convert.ToInt32(Console.ReadLine());

for (int i = start; i <= end; i++) {
    int num = i;
    int originalNum = num;
    int sum = 0;

    while (num > 0) {
        sum += num % 10;
        num /= 10;
    }

    if (originalNum % sum == 0)
        Console.WriteLine(originalNum + " is a Harshad Number");
    }
}

```

5. Print the Pattern of a Hollow Diamond:

```

using System;

class HollowDiamondPattern {
    static void Main() {
        Console.WriteLine("Enter the number of rows: ");
        int rows = Convert.ToInt32(Console.ReadLine());

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows - i; j++) {
                Console.Write(" ");
            }
        }
    }
}

```



```

    }
    for (int k = 1; k <= 2 * i - 1; k++) {
        if (k == 1 || k == 2 * i - 1)
            Console.Write("*");
        else
            Console.Write(" ");
    }
    Console.WriteLine();
}

for (int i = rows - 1; i >= 1; i--) {
    for (int j = 1; j <= rows - i; j++) {
        Console.Write(" ");
    }
    for (int k = 1; k <= 2 * i - 1; k++) {
        if (k == 1 || k == 2 * i - 1)
            Console.Write("*");
        else
            Console.Write(" ");
    }
    Console.WriteLine();
}
}
}

```

6. Check for Keith Number:

using System;

```

class KeithNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
    }
}

```

```

string input = Console.ReadLine();
int num = int.Parse(input);
int[] arr = new int[input.Length];

for (int i = 0; i < input.Length; i++) {
    arr[i] = input[i] - '0';
}

int sum = 0;
while (sum < num) {
    sum = 0;
    for (int i = 0; i < input.Length; i++) {
        sum += arr[i];
        if (i < input.Length - 1) {
            arr[i] = arr[i + 1];
        }
        else {
            arr[i] = sum;
        }
    }
    if (sum == num) {
        Console.WriteLine("Keith Number");
        return;
    }
}
Console.WriteLine("Not a Keith Number");
}
}

```

7. Calculate the Power of a Number Using Recursion:

using System;

```

class PowerUsingRecursion {
    static int Power(int x, int y) {
        if (y == 0)
            return 1;
        else
            return x * Power(x, y - 1);
    }

    static void Main() {
        Console.WriteLine("Enter the base: ");
        int baseNum = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the exponent: ");
        int exponent = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Result: " + Power(baseNum, exponent));
    }
}

```

8. Check for Amicable Numbers:

```
using System;
```

```

class AmicableNumbers {
    static int GetSumOfDivisors(int num) {
        int sum = 0;
        for (int i = 1; i <= num / 2; i++) {
            if (num % i == 0) {

                sum += i;
            }
        }
        return sum;
    }
}

```

```

    }

    static void Main() {
        Console.WriteLine("Enter the first number: ");
        int num1 = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the second number: ");
        int num2 = Convert.ToInt32(Console.ReadLine());

        int sum1 = GetSumOfDivisors(num1);
        int sum2 = GetSumOfDivisors(num2);

        if (sum1 == num2 && sum2 == num1)
            Console.WriteLine("Amicable Numbers");
        else
            Console.WriteLine("Not Amicable Numbers");
    }
}

```

9. Find the GCD and LCM of Two Numbers:

using System;

```

class GCDandLCM {
    static int FindGCD(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }

    static int FindLCM(int a, int b) {

```

```

        return (a * b) / FindGCD(a, b);
    }

    static void Main() {
        Console.WriteLine("Enter the first number: ");
        int num1 = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the second number: ");
        int num2 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("GCD: " + FindGCD(num1, num2));
        Console.WriteLine("LCM: " + FindLCM(num1, num2));
    }
}

```

10. Check for Keith Number in a Range:

using System;

```

class KeithNumbersInRange {
    static void Main() {
        Console.WriteLine("Enter the starting number: ");
        int start = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the ending number: ");
        int end = Convert.ToInt32(Console.ReadLine());

        for (int i = start; i <= end; i++) {
            string input = i.ToString();
            int num = int.Parse(input);
            int[] arr = new int[input.Length];

            for (int j = 0; j < input.Length; j++) {
                arr[j] = input[j] - '0';
            }

```

```

int sum = 0;
while (sum < num) {
    sum = 0;
    for (int j = 0; j < input.Length; j++) {
        sum += arr[j];
        if (j < input.Length - 1) {
            arr[j] = arr[j + 1];
        }
        else {
            arr[j] = sum;
        }
    }
    if (sum == num) {
        Console.WriteLine(num + " is a Keith Number");
        break;
    }
}
}
}
}
}
}
}
}

```

SET – 8

1. Check for Kaprekar Number:

```

using System;

class KaprekarNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
    }
}

```

```

int square = num * num;
string strSquare = square.ToString();

for (int i = 1; i < strSquare.Length; i++) {
    int part1 = int.Parse(strSquare.Substring(0, i));
    int part2 = int.Parse(strSquare.Substring(i));

    if (part1 + part2 == num) {
        Console.WriteLine("Kaprekar Number");
        return;
    }
}
Console.WriteLine("Not a Kaprekar Number");
}
}

```

2. Implement Bubble Sort:

```

using System;

class BubbleSort {
    static void Main() {
        int[] arr = { 64, 34, 25, 12, 22, 11, 90 };
        int n = arr.Length;

        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
}

```

```

    }

    Console.WriteLine("Sorted array: ");
    foreach (int element in arr) {
        Console.WriteLine(element + " ");
    }
}
}

```

3. Check for Perfect Number:

using System;

```

class PerfectNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int sum = 0;

        for (int i = 1; i < num; i++) {
            if (num % i == 0) {
                sum += i;
            }
        }

        if (sum == num)
            Console.WriteLine("Perfect Number");
        else
            Console.WriteLine("Not a Perfect Number");
    }
}

```


4. Find the Factors of a Number:

using System;

```
class Factors {  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
        Console.Write("Factors of " + num + " are: ");  
  
        for (int i = 1; i <= num; i++) {  
            if (num % i == 0) {  
                Console.Write(i + " ");  
            }  
        }  
    }  
}
```

5. Print the Pattern of a Right Triangle:

using System;

```
class RightTrianglePattern {  
    static void Main() {  
        Console.WriteLine("Enter the number of rows: ");  
        int rows = Convert.ToInt32(Console.ReadLine());  
  
        for (int i = 1; i <= rows; i++) {  
            for (int j = 1; j <= i; j++) {  
                Console.Write("*");  
            }  
            Console.WriteLine();  
        }  
    }  
}
```

```
}  
}
```

6. Check for Prime Number:

using System;

```
class PrimeNumber {  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
        bool isPrime = true;  
  
        if (num == 1 || num == 0) {  
            isPrime = false;  
        }  
        else {  
            for (int i = 2; i <= Math.Sqrt(num); i++) {  
                if (num % i == 0) {  
                    isPrime = false;  
                    break;  
                }  
            }  
        }  
  
        if (isPrime)  
            Console.WriteLine("Prime Number");  
        else  
            Console.WriteLine("Not a Prime Number");  
    }  
}
```

7. Calculate the Sum of Natural Numbers:

using System;

```
class SumOfNaturalNumbers {  
    static void Main() {  
        Console.WriteLine("Enter the value of n: ");  
        int n = Convert.ToInt32(Console.ReadLine());  
  
        int sum = 0;  
        for (int i = 1; i <= n; i++) {  
            sum += i;  
        }  
  
        Console.WriteLine("Sum of first " + n + " natural numbers: " +  
sum);  
    }  
}
```

8. Print the Pattern of a Right Triangle using Numbers:

using System;

```
class NumberRightTrianglePattern {  
    static void Main() {  
        Console.WriteLine("Enter the number of rows: ");  
        int rows = Convert.ToInt32(Console.ReadLine());  
  
        for (int i = 1; i <= rows; i++) {  
            for (int j = 1; j <= i; j++) {  
                Console.Write(j);  
            }  
            Console.WriteLine();  
        }  
    }  
}
```

```
}  
}
```

9. Check for Composite Number:

using System;

```
class CompositeNumber {  
    static void Main() {  
        Console.WriteLine("Enter a number: ");  
        int num = Convert.ToInt32(Console.ReadLine());  
        bool isComposite = false;  
  
        if (num == 0 || num == 1) {  
            isComposite = false;  
        }  
        else {  
            for (int i = 2; i <= num / 2; i++) {  
                if (num % i == 0) {  
                    isComposite = true;  
                    break;  
                }  
            }  
        }  
  
        if (isComposite)  
            Console.WriteLine("Composite Number");  
        else  
            Console.WriteLine("Not a Composite Number");  
    }  
}
```

10. Check for Palindrome String:

using System;

```
class PalindromeString {
    static void Main() {
        Console.WriteLine("Enter a string: ");
        string str = Console.ReadLine();
        string reversedStr = "";

        for (int i = str.Length - 1; i >= 0; i--) {
            reversedStr += str[i];
        }

        if (str.Equals(reversedStr, StringComparison.OrdinalIgnoreCase))
            Console.WriteLine("Palindrome String");
        else
            Console.WriteLine("Not a Palindrome String");
    }
}
```

SET – 9

1. Check for Armstrong Number:

using System;

```
class ArmstrongNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int originalNum = num;
        int sum = 0;
```

```

int digits = num.ToString().Length;

while (num > 0) {
    int digit = num % 10;
    sum += (int)Math.Pow(digit, digits);
    num /= 10;
}

if (sum == originalNum)
    Console.WriteLine("Armstrong Number");
else
    Console.WriteLine("Not an Armstrong Number");
}
}

```

2. Implement Selection Sort:

```

using System;

class SelectionSort {
    static void Main() {
        int[] arr = { 64, 25, 12, 22, 11 };
        int n = arr.Length;

        for (int i = 0; i < n - 1; i++) {
            int minIndex = i;
            for (int j = i + 1; j < n; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }

            int temp = arr[minIndex];

```

```

        arr[minIndex] = arr[i];
        arr[i] = temp;
    }

    Console.WriteLine("Sorted array: ");
    foreach (int element in arr) {
        Console.Write(element + " ");
    }
}

```

3. Check for Disarium Number:

```

using System;

class DisariumNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int originalNum = num;
        string numString = num.ToString();
        int sum = 0;

        for (int i = 0; i < numString.Length; i++) {
            sum += (int)Math.Pow(int.Parse(numString[i].ToString()), i +
1);
        }

        if (sum == originalNum)
            Console.WriteLine("Disarium Number");
        else
            Console.WriteLine("Not a Disarium Number");
    }
}

```

```
}
```

4. Find the Sum of Natural Numbers Using Recursion:

```
using System;
```

```
class SumOfNaturalNumbersUsingRecursion {  
    static int Sum(int n) {  
        if (n == 0)  
            return 0;  
        else  
            return n + Sum(n - 1);  
    }  
  
    static void Main() {  
        Console.WriteLine("Enter the value of n: ");  
        int n = Convert.ToInt32(Console.ReadLine());  
  
        int sum = Sum(n);  
        Console.WriteLine("Sum of first " + n + " natural numbers: " +  
sum);  
    }  
}
```

5. Print the Pattern of a Mirrored Right Triangle:

```
using System;
```

```
class MirroredRightTrianglePattern {  
    static void Main() {  
        Console.WriteLine("Enter the number of rows: ");  
        int rows = Convert.ToInt32(Console.ReadLine());
```



```

        for (int i = 1; i <= rows; i++) {
            for (int j = 1; j <= rows - i; j++) {
                Console.Write(" ");
            }
            for (int k = 1; k <= i; k++) {
                Console.Write("*");
            }
            Console.WriteLine();
        }
    }
}

```

6. Check for Perfect Number in a Range:

using System;

```

class PerfectNumbersInRange {
    static void Main() {
        Console.WriteLine("Enter the starting number: ");
        int start = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the ending number: ");
        int end = Convert.ToInt32(Console.ReadLine());

        for (int i = start; i <= end; i++) {
            int num = i;
            int sum = 0;

            for (int j = 1; j < num; j++) {
                if (num % j == 0) {
                    sum += j;
                }
            }

            }
        }
    }
}

```

```

        if (sum == num)
            Console.WriteLine(num + " is a Perfect Number");
    }
}
}

```

7. Calculate the Factorial of a Number Using Recursion:

using System;

```

class FactorialUsingRecursion {
    static int Factorial(int n) {
        if (n == 0)
            return 1;
        else
            return n * Factorial(n - 1);
    }

    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());

        int factorial = Factorial(num);
        Console.WriteLine("Factorial: " + factorial);
    }
}

```

8. Check for Deficient Number:

using System;

```

class DeficientNumber {
    static void Main() {

```

```

Console.WriteLine("Enter a number: ");
int num = Convert.ToInt32(Console.ReadLine());
int sum = 0;

for (int i = 1; i <= num / 2; i++) {
    if (num % i == 0)
        sum += i;
}

if (sum < num)
    Console.WriteLine("Deficient Number");
else
    Console.WriteLine("Not a Deficient Number");
}
}

```

9. Find the Fibonacci Series Up to a Given Number:

using System;

```

class FibonacciSeries {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int a = 0, b = 1, c = 0;

        Console.Write("Fibonacci Series: " + a + " " + b + " ");

        for (int i = 2; c < num; i++) {
            c = a + b;
            if (c <= num)
                Console.Write(c + " ");
            a = b;

```

```

        b = c;
    }
}
}

```

10. Check for Neon Number:

using System;

```

class NeonNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int square = num * num;
        int sum = 0;

        while (square != 0) {
            sum += square % 10;
            square /= 10;
        }

        if (sum == num)
            Console.WriteLine("Neon Number");
        else
            Console.WriteLine("Not a Neon Number");
    }
}

```

SET – 10

1. Check for Happy Number in a Range:

```
using System;
```

```
class HappyNumbersInRange {  
    static int SquareSumOfDigits(int num) {  
        int sum = 0;  
        while (num > 0) {  
            int digit = num % 10;  
            sum += digit * digit;  
            num /= 10;  
        }  
        return sum;  
    }  
  
    static bool IsHappyNumber(int num) {  
        while (num != 1 && num != 4) {  
            num = SquareSumOfDigits(num);  
        }  
        return num == 1;  
    }  
  
    static void Main() {  
        Console.WriteLine("Enter the starting number: ");  
        int start = Convert.ToInt32(Console.ReadLine());  
  
        Console.WriteLine("Enter the ending number: ");  
        int end = Convert.ToInt32(Console.ReadLine());  
  
        for (int i = start; i <= end; i++) {  
            if (IsHappyNumber(i)) {  
                Console.WriteLine(i + " is a Happy Number");  
            }  
        }  
    }  
}
```

```
}
```

2. Implement Insertion Sort:

```
using System;
```

```
class InsertionSort {  
    static void Main() {  
        int[] arr = { 12, 11, 13, 5, 6 };  
        int n = arr.Length;  
  
        for (int i = 1; i < n; i++) {  
            int key = arr[i];  
            int j = i - 1;  
  
            while (j >= 0 && arr[j] > key) {  
                arr[j + 1] = arr[j];  
                j = j - 1;  
            }  
            arr[j + 1] = key;  
        }  
  
        Console.WriteLine("Sorted array: ");  
        foreach (int element in arr) {  
            Console.Write(element + " ");  
        }  
    }  
}
```

3. Check for Pronic Number:

```
using System;
```

```

class PronicNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int flag = 0;

        for (int i = 0; i < num; i++) {
            if (i * (i + 1) == num) {
                flag = 1;
                break;
            }
        }

        if (flag == 1)
            Console.WriteLine("Pronic Number");
        else
            Console.WriteLine("Not a Pronic Number");
    }
}

```

4. Find the LCM of N Numbers:

using System;

```

class LCMofNNumbers {
    static int FindGCD(int a, int b) {
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        return a;
    }
}

```

```

    }

    static int FindLCM(int[] arr, int n) {
        int lcm = arr[0];
        for (int i = 1; i < n; i++) {
            lcm = (lcm * arr[i]) / FindGCD(lcm, arr[i]);
        }
        return lcm;
    }

    static void Main() {
        Console.WriteLine("Enter the value of n: ");
        int n = Convert.ToInt32(Console.ReadLine());
        int[] arr = new int[n];

        for (int i = 0; i < n; i++) {
            Console.WriteLine("Enter number " + (i + 1) + ": ");
            arr[i] = Convert.ToInt32(Console.ReadLine());
        }

        Console.WriteLine("LCM: " + FindLCM(arr, n));
    }
}

```

5. Print the Pattern of a Inverted Pyramid:

using System;

```

class InvertedPyramidPattern {
    static void Main() {
        Console.WriteLine("Enter the number of rows: ");
        int rows = Convert.ToInt32(Console.ReadLine());

        for (int i = rows; i >= 1; i--) {

```



```

        for (int j = 1; j <= rows - i; j++) {
            Console.Write(" ");
        }
        for (int k = 1; k <= 2 * i - 1; k++) {
            Console.Write("*");
        }
        Console.WriteLine();
    }
}

```

6. Check for Strong Number:

using System;

```

class StrongNumber {
    static int Factorial(int num) {
        if (num == 0 || num == 1)
            return 1;
        return num * Factorial(num - 1);
    }

    static void Main() {
        Console.WriteLine("Enter a number: ");
        int num = Convert.ToInt32(Console.ReadLine());
        int originalNum = num;
        int sum = 0;

        while (num > 0) {
            int digit = num % 10;
            sum += Factorial(digit);
            num /= 10;
        }
    }
}

```

```

    }

    if (sum == originalNum)
        Console.WriteLine("Strong Number");
    else
        Console.WriteLine("Not a Strong Number");
    }
}

```

7. Calculate the Power of a Number Using Iteration:

using System;

```

class PowerUsingIteration {
    static int Power(int x, int y) {
        int result = 1;
        for (int i = 1; i <= y; i++) {
            result *= x;
        }
        return result;
    }

    static void Main() {
        Console.WriteLine("Enter the base: ");
        int baseNum = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter the exponent: ");
        int exponent = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Result: " + Power(baseNum, exponent));
    }
}

```

8. Check for Duck Number:

using System;

```
class DuckNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        string num = Console.ReadLine();

        if (num.Contains("0") && num[0] != '0')
            Console.WriteLine("Duck Number");
        else
            Console.WriteLine("Not a Duck Number");
    }
}
```

9. Check for Emirp Number:

using System;

```
class EmirpNumber {
    static bool IsPrime(int num) {
        if (num == 1 || num == 0) {
            return false;
        }
        for (int i = 2; i <= Math.Sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

```

static void Main() {
    Console.WriteLine("Enter a number: ");
    int num = Convert.ToInt32(Console.ReadLine());
    int reverse = 0;
    int temp = num;

    while (temp != 0) {
        int digit = temp % 10;
        reverse = reverse * 10 + digit;
        temp /= 10;
    }

    if (IsPrime(num) && IsPrime(reverse) && num != reverse)
        Console.WriteLine("Emirp Number");
    else
        Console.WriteLine("Not an Emirp Number");
    }
}

```

10. Check for Strobogrammatic Number:

```

using System;

class StrobogrammaticNumber {
    static void Main() {
        Console.WriteLine("Enter a number: ");
        string num = Console.ReadLine();
        string validChars = "01689";

        for (int i = 0; i < (num.Length + 1) / 2; i++) {
            if (!validChars.Contains(num[i]) || num[i] != num[num.Length
- 1 - i]) {

```

```
        Console.WriteLine("Not a Strobogrammatic Number");  
        return;  
    }  
}  
  
    Console.WriteLine("Strobogrammatic Number");  
}  
}
```

C#:

C# is a powerful programming language that can be used to create a wide variety of applications, ranging from desktop applications to web services and games. Some common types of applications that can be developed using C# include:

1. **Desktop Applications:** C# can be used to create robust and feature-rich desktop applications for Windows, including business applications, utilities, tools, and more. Windows Forms and Windows Presentation Foundation (WPF) are popular frameworks for building desktop applications using C#.
2. **Web Applications:** C# is extensively used for building web applications, including dynamic websites, web services, and web APIs. ASP.NET, ASP.NET Core, and Blazor are popular frameworks that allow developers to create web applications using C#.
3. **Mobile Applications:** With the help of cross-platform development frameworks like Xamarin, C# can be used to create cross-platform mobile applications for iOS and Android. Xamarin allows developers to share code across multiple platforms, streamlining the development process.
4. **Games:** C# is widely used in the game development industry. Game engines such as Unity and Unreal Engine offer extensive support for C#, making it a popular choice for developing 2D and 3D games across various platforms, including PC, consoles, and mobile devices.
5. **Cloud Applications:** C# can be used to build cloud applications and services on popular cloud platforms such as Microsoft Azure. It enables the development of scalable, reliable, and high-performance cloud-based applications and services.

6. IoT (Internet of Things) Applications: C# can be used in IoT applications to develop software for embedded devices, IoT gateways, and IoT solutions. .NET Core supports cross-platform development, making it suitable for creating IoT applications that run on various devices and platforms.

7. Machine Learning and AI Applications: C# can be used in developing applications related to machine learning and artificial intelligence, leveraging frameworks such as ML.NET. It allows developers to create and deploy machine learning models and integrate them into various applications.

8. Database Applications: C# can be used to build database applications, including data-driven applications, data analysis tools, and database management systems. It integrates seamlessly with various database technologies, such as Microsoft SQL Server, MySQL, and PostgreSQL.

9. Enterprise Applications: C# is often used for developing enterprise-level applications, including ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), and other business management systems. It enables the creation of scalable, secure, and robust applications to meet the complex needs of businesses.

10. Web Services and APIs: C# can be used to create web services and APIs that facilitate communication and data exchange between different software applications. RESTful APIs and SOAP-based services can be developed using C# to enable interoperability and integration between various systems.

These are just a few examples of the wide range of applications that can be developed using C#. With its versatility, extensive framework support, and a rich set of libraries, C# is a preferred choice for many developers across different domains.

HOW TO RUN C# PROGRAM

To run a C# program, you'll need an integrated development environment (IDE) like Visual Studio or Visual Studio Code. Here's a basic guide to running a C# program using Visual Studio:

1. Install Visual Studio:

If you haven't already, download and install Visual Studio from the official Microsoft website.

2. Create a new C# project:

Open Visual Studio and select "Create a new project." Choose the appropriate project template depending on your requirements. For a simple console application, choose the "Console App (.NET)" template.

3. Write your C# code:

In the newly created project, you'll find a file named `Program.cs`. Open it and write your C# code in the `Main` method or any other appropriate methods.

4. Build the project:

After writing the code, build the project by selecting the "Build" option from the menu.

5. Run the program:

Once the project is built successfully, you can run the program by clicking the "Start" button or by pressing the "F5" key. This will run your C# program, and if it's a console application, you'll see the output in the console window.

If you prefer a simpler approach without using an IDE, you can also use the .NET Core command-line interface (CLI). Here's a basic guide to running a C# program using the .NET CLI:

1. Install .NET SDK:

If you haven't already, download and install the .NET SDK from the official .NET website.

2. Create a new folder for your C# program:

Open your preferred terminal or command prompt and create a new folder for your C# program.

3. Write your C# code:

Inside the new folder, create a file with a `.cs` extension (e.g., `program.cs`) and write your C# code.

4. Compile and run the program:

In the terminal or command prompt, navigate to the folder containing your C# file. Use the following commands to compile and run your program:

dotnet new console # Create a new console application

dotnet run # Compile and run the program

This will compile and run your C# program, and you'll see the output in the terminal or command prompt.

These are the basic steps for running a C# program using Visual Studio or the .NET CLI.

Thank You

Table of Contents

C#	1
C#:	77
How to Run C# Program	79